UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/796,317	03/09/2004	**Shawn A.P. Smith**	T00107	2095

33438 7590 09/02/2009
HAMILTON & TERRILE, LLP
P.O. BOX 203518
AUSTIN, TX 78720

EXAMINER	
HWA, SHYUE JIUNN	

ART UNIT	PAPER NUMBER
2163	

NOTIFICATION DATE	DELIVERY MODE
09/02/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

tmunoz@hamiltonterrile.com

Office Action Summary	**Application No.** 10/796,317	**Applicant(s)** SMITH ET AL.	
	Examiner JAMES HWA	**Art Unit** 2163	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 May 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 are pending in this office action. No claims are amended or canceled. This action is responsive to Applicant's application filed May 26, 2009.

Response to Arguments

2. Applicant's arguments filed 05/26/2009 have been fully considered but are not persuasive.

Applicant argued that, the cited arts fail to teach or suggest the limitation "processing each entry in the memory to identify entries in the subset of log file entries that belong to a complete client session" and "grouping entries in the subset that belong to a complete client session" in claim 1, "for each identified user session, an index to identify all records in the ring buffer that are associated with the identified user session and to identify all start or end records" and processing "the index to group all records in the ring buffer belonging to a complete user session" in claim 8, "a processing engine to process a subset of the plurality of server request entries to group the server request entries by session using the session identifier in each server request entry" in claim 11, "means for processing the subset of the network session data to group said network session data by session" and "means for generating a first output file containing network session data grouped by session" in claim 18.

Examiner respectfully disagree all allegations as argued. Examiner, in her previous office action gave detail explanation of claimed limitation and pointed out exact locations in the cited prior art. Examiner is entitled to give claim limitations their broadest reasonable interpretation in light of the specification. See. MPEP 2111[R-1]

Interpretation of Claims-Broadest Reasonable Interpretation During patent examination, the pending claims must be 'given the broadest reasonable interpretation consistent with the specification'. Applicant always has the opportunity to amend the claims during prosecution and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 162 USPW 541,550-51 (CCPA 1969).

In response to applicant's argument, Nareddy teaches the routine processes each entry in the log file based on the parser configuration data. The routine filters out certain log entries, normalizes the attribute values (e.g., based on time range, user IP address, success code or corrupt status; also see elements 602, 603, 605, 607 of figure 6) of the log entries, and generates entries in the dimension tables for the attributes of the log entries. After processing all the log entries, the parser identifies user sessions (e.g., complete client session) and generates various statistics. The routine invokes the generate dimensions routine to update the dimension tables based on the selected log entry and to add an entry into the log entry fact table. The routine updates the statistics for the log file. For example, the routine may track the number of log entries that have been filtered out. The routine then loops to select the next log entry, the routine outputs the log file statistics, then the routine invokes the identify sessions routine that scans the log entry table to identify the user sessions (e.g., client sessions) and updates a session dimension table and generate aggregate statistics routine to generate various statistics and then completes. The filter log entry routine is passed a log entry and determines whether the log entry should be filtered out. The routine determines (e.g., by client session or field corrupt; see also element 607 of figure 6) whether the filter out conditions has been satisfied. In decision block, the routine determines whether the log entry has a field count problem. A field count problem arises when the number of fields in the log entry does not correspond to the number of expected fields for that log entry (column 13, lines 10-60). The retrieved category type definition information may already be separated into separate version groups (e.g., grouping entries). If it is possible to determine from the information received that a subset of the version groups will apply to all of the log entries (e.g., identify all records) in the log file, the routine could discard the definitions that are not in those version groups (column 49, lines 33-58).

The data processor component inputs parser configuration data and a log file and updates the main data warehouse. The parser configuration data may include a mapping of actual web sites to logical sites and a mapping of a combination of Uniform Resource Identifier (URL) and query strings of the log entries to page definitions (e.g., categories) and event definitions. The generate dimensions component identifies the various dimensions related to a log entry. For example, a dimension may be the Uniform Resource Identifier of the entry or the logical site identifier. The identify sessions component (e.g., session identifier; see also element 314 of figure 3) processes the parsed log file data stored in the local data warehouse to identify user sessions. A user session generally refers to the concept of a series of web page accesses that may be related in some way, such as by temporal proximity. The generate aggregate statistics component aggregates data for the log file being processed as each log entry is processed or after the log file is parsed (column 8, lines

26-60). Each entry in a dimension table includes the attribute value (e.g., user identifier) and a hash value. The hash value may be used by the loader when transferring information to the main data warehouse. Also, each entry has a local identifier, which may be an index into the local dimension table. The loader maps these local identifiers to their corresponding main identifiers that are used in the main data warehouse (column 14, lines 53-64; see also 802 of figure 8).

When the data warehouse server receives customer data, it converts the customer data into a format that is more conducive to processing by decision support system applications used to analyze customer data. For example, the data warehouse server may analyze low-level navigation events (e.g., each HTTP request that is received by the customer web site) to identify high-level events (e.g., a user session). The data warehouse server then stores the converted data into a data warehouse (column 5, lines 53-60; element 603 of figure 6, filter by IP address (e.g., group the server request by server IP address); see also column 25 table 3, server ID=IP address 209.67.55.102).

A user can request a particular resource (e.g., a web page or a file) that is available from a web server by specifying a unique URI for that resource. A URI can be a URL, Uniform Resource Name (URN), or any other formatted string that identifies a network resource (column 1, lines 59-67). The data warehouse may include fact tables and dimension tables that represent high-level facts and attributes derived from the low-level facts and attributes of the log files. For example, the higher level category of a web page may be identified using a mapping of web page URIs to categories. These categories may be stored in a category dimension table. Also, certain facts, such as the collection of log entries that comprise a single user web access session or visit (e.g., group by network session), may only be derivable by analyzing a series of log entries (column 6, lines 25-36).

As to “identified user session and to identify all start or end records” and "grouping entries in the subset that belong to a complete client session", Nareddy teaches many of the data parsing information entries include values for beginning and ending dates that define an effective date range for which the data parsing information is valid (column 42, lines 37-50; see also column 12, table 2, Event Definition=Login (e.g., start record), Logout (e.g., end record), Abort, Forgot (e.g., incomplete session), View New of the user session; see also elements 602, 607 of figure 6, filter corrupt =True or False).

Although Nareddy teaches the routine creates a mapping of the local identifier (index into the local dimension table) of the selected entry to the main identifier (index into the main dimension table) for that selected entry (column 17, line 65 to column 18, line 3).

Nareddy does not explicitly teach the claimed limitation "an index to identify all records in the ring buffer” and processing "the index to group all records in the ring buffer".

Carothers teaches the called method invokes an isolating the next transaction journal message API to position the next transaction message in the transaction journal log, read the transaction message contents, and insert the transaction message in a static buffer (e.g. ring buffer) (page 2, paragraph 0014).

Also, Dow teaches the event buffer module receives and aggregates events from an event index. In one embodiment, the event buffer receives selected events from the content navigator module that it has read from an index file. The event buffer module stores a plurality of events for analysis. In one embodiment, the event buffer module buffers event data for a set period of time, such as 2 minutes. For example, the event buffer module may include a 2-minute ring buffer. In an alternate embodiment, the event buffer module buffers event data for an entire video program. In still another embodiment, the event buffer module stores a set number of events before the earliest stored event data is pushed off the event stack to make room for a new event. The other modules in the group detector module analyze the data stored by the event buffer module (page 14, paragraph 0112).

Applicant argued that it would not be obvious to combine the two references and no motivation to combine.

In response to applicant's argument, The examiner respectfully submits that to establish a prima facie case of obviousness under 35 USC 103, references must provide motivation or suggestion either in the references themselves, or in the knowledge generally available to one of ordinary skill in the art; must be analogous; and must teach all the claimed limitations.

In this case, the instant application is concerned to a system and method are provided for grouping and processing log file entries using a memory window to select a subset of the log file records for identifying complete session records that may be analyzed or parsed.

As discussed in the office action, Nareddy provides a data warehouse system collects customer data from the customer web sites and stores the data at a data warehouse server. The customer data may include application event data (e.g., click stream log files); user attributes data of users of the customer web site, product data, shopping cart data and so on (column 5, lines 28-35).

Similarly, Carothers teaches is related to a method manage information concerning financial transactions, such as home banking system transactions and automatic teller machine transactions (page 1, paragraph 0002). Dow provides systems and methods for recording, indexing, and navigating video content.

Importantly, Carothers provide a system and method for managing information concerning financial transactions that takes existing messages and the existing software for home banking and develops messages which go into an MIS database and easily and efficiently provides many different types of reports (page 1, paragraph 0004).

Dow utilize a content-based index that defines segments within a video presentation to improved navigation of video content recorded from a television broadcast and allow a user to automatically view commercials from a stored video presentation without the intervening program content (page 2, paragraph 0011).

As discussed above, a person of an ordinary skill in the art at the time the invention was made would recognize the advantage of Carothers and Dow to add the Carothers and Dow's teaching of using index to identify all records in the ring buffer to

Nareddy's system direct processed event data to the navigation function module allows rapid, non-sequential access to the data stored (page 9, paragraph 0082).

Therefore, the 103 rejection for claims is proper and make the record clear.

Remarks

3. The inventive concept attempts to improve on system and method provided for grouping and processing log file entries using a memory window to select a subset of the log file records for identifying complete session records. However, this is a conventional concept found in Nareddy's prior art.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

4. Claims 1-7, 11 and 13-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nareddy et al. (US Patent No. 7,107,338 B1, hereinafter "Nareddy") in view of in view of Carothers et al. (US Patent Application No. 2002/0016771 A1, hereinafter "Carothers").

As to claim 1, Nareddy teaches the claimed limitations:

"A method for grouping log file entries by session" as a method and system for providing customers with access to and analysis of interaction or usage data (e.g., navigation data collected at customer web sites or computer program event information) is provided. The data warehouse server may analyze low-level navigation events (e.g., each HTTP request that is received by the customer web site) to identify high-level events (e.g., a user session) (column 1, lines 27-60).

"Storing a log file of entries in a memory, each of said entries identifying a client request to a server" as the interaction or usage data, hereinafter interaction data or event data, may be stored in the log files and supplemented with data from other sources, such as product databases and customer invoices (column 1, lines 27-60).

"Retrieving a subset of log file entries from the memory" as the retrieved category type definition information may already be separated into separate version groups. If it is possible to determine from the information received that a subset of the version groups will apply to all of the log entries in the log file, the routine could discard (or not initially retrieve) the definitions that are not in those version groups (column 49, lines 33-58).

"Processing each entry in the memory to identify entries in the subset of log file entries that belong to a complete client session" as the parse log data routine that implements the main routine of parser in one embodiment. The routine processes each entry in the log file based on the parser configuration data. The routine filters out certain log entries, normalizes the attribute values of the log entries, and generates entries in the dimension tables for the attributes of the log entries. After processing all the log entries, the parser identifies user sessions and generates various statistics. The routine invokes the generate dimensions routine to update the dimension tables based on the selected log entry and to add an entry into the log entry fact table. The routine updates the statistics for the log file. For example, the routine may track the number of log entries that have been filtered out. The routine then loops to select the next log entry, the routine outputs the log file statistics, then the routine invokes the identify sessions routine that scans the log entry table to identify the user sessions and updates a session dimension table and generate aggregate statistics routine to generate various statistics and then completes. The filter log entry routine is passed a log entry and determines whether the log entry should be filtered out. The routine determines whether the filter out conditions has been satisfied. In decision block, the routine determines whether the log entry has a field count problem. A field count problem arises when the number of fields in the log entry does not correspond to the number of expected fields for that log entry (column 13, lines 10-60; see also element 607 of figure 6, true or false of corrupt). If multiple definitions are allowed or if the selected definition does not match the log entry, the routine continues to determine if there are more category type definitions in the selected version group. If multiple definitions are not allowed per log entry, the routine instead continues determine if there are more log entries to be processed. If so, the routine returns to select the next log entry for processing, If there are more log files, the routine continues, and if not then the routine continues and ends. The retrieved category type definition information may already be separated into separate version groups. If it is possible to determine from the information received a subset of the version groups will apply to the entire log entries in the log file, the routine could discard the definitions that are not in those version groups (column 49, lines 9-58).

"Entries in the subset that belong to a complete client session" as teaches the routine processes each entry in the log file based on the parser configuration data. The routine filters out certain log entries, normalizes the attribute values (e.g., based on time range, user IP address, success code or corrupt status; also see elements 602, 603, 605, 607 of figure 6) of the log entries, and generates entries in the dimension tables for the attributes of the log entries. After processing all the log entries, the parser identifies user sessions (e.g., client session) and generates various statistics. The routine then loops to select the next log entry, the routine outputs the log file statistics, then the routine invokes the identify sessions routine that scans the log entry table to identify the user sessions and updates a session dimension table and generate aggregate statistics routine to generate various statistics (column 13, lines 10-60). The retrieved category type definition information may already be separated into separate version groups. If it is possible to determine from the information received that a subset of the version groups will apply to all of the log entries in the log file, the routine could discard (or not initially retrieve) the definitions that are not in those version groups (column 49, lines 33-58).

Nareddy does not explicitly teach the claimed limitation "grouping" entries in the subset that belong to a complete client session.

Carothers teaches each transaction entry is examined to determine a type of function for the transaction, each transaction encountered in the examination is written as a line to an output file, and all transactions encountered are grouped by transaction according to a data column of the transaction journal record that refers back to a session log record. The unique integer key value is assigned to each individual transaction record in the ASCII text output files and written to a pre-defined column in each output file for loading to the relational database system. In addition, transaction times are computed for each of the individual transaction records and added to the corresponding output file (page 1, paragraph 0010; see also element 120 of figure 15, Session completion details Report). Netscape sessions is the number of valid sessions where the customer used a Netscape browser; IE sessions is the number of valid sessions where the customer used an Internet Explorer browser (page 9, paragraph 0119; see also element 116 of figure 16).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy and Carothers before him/her, to modify Nareddy grouping entries in the subset that belong to a complete client session because that would provide a method for managing information concerning financial transactions for home banking and develops messages which go into the database easily and efficiently as taught by Carothers (page 1, paragraph 0004).

As to claim 2, Nareddy teaches the claimed limitations:

“A complete client session is identified by identifying all entries in the subset that are associated with a particular client session and that include both a beginning entry and an end entry” as many of the data parsing information entries include values for beginning and ending dates that define an effective date range for which the data parsing information is valid (column 42, lines 37-50; see also column 12, table 2, Event Definition=Login (e.g., start record), Logout (e.g., end record), Abort, Forgot (e.g., incomplete session), View New of the user session; see also elements 602, 607 of figure 6, filter corrupt =True or False).

As to claim 3, Nareddy teaches the claimed limitations:

“An end entry is identified as any entry that corresponds to a logout request” as Table 2 is an example portion of parser configuration data. The event type definitions map a site identifier, URI pattern, and query string pattern to an event type and value. EVENTDEFINITION= Logout, Logout, 1, [prefix]=/registration/logout.asp,

Also, Carothers teaches in order to minimize the effect of customer time-outs, for valid sessions that end due to a customer timeout, the session length of each of those sessions is reduced by a predetermined time (page 9, paragraph 0120; see also figure 16). Sessions w/timeout-percent of all is the number of valid customer sessions that ended due to a session time out as a percentage of all valid sessions; sessions w/other end-count is the number of valid customer sessions that ended due to some reason other than a customer sign-off or timeout (page 10, paragraph 0125).

As to claim 4, Nareddy teaches the claimed limitations:

"An end entry for a client session is identified as any entry associated with that client session that has no other entries for that client session that occur within a session expiration window" as the identify sessions component processes the parsed log file data stored in the local data warehouse to identify user sessions. A user session generally refers to the concept of a series of web page accesses that may be related in some way, such as by temporal proximity (column 8, lines 52-60). This web page is shown displayed within a web browser display window. The displayed web page includes multiple frames that are each able to display different content, including a control frame with various user-selectable controls and display frames in which customer-specific information is displayed. In the illustrated embodiment, the URL indication corresponds to the information displayed in the display frames. The path portion of the indicated URL specifies an executable Active Server Page program on the server that will supply the content displayed in the display frames, and the indicated URL also includes a query string portion that will be supplied as input to the executable program (column 21, lines 30-40).

As to claim 5, Nareddy teaches the claimed limitation:

"an end entry for a client session is identified as any entry having a first timestamp value, where the difference between first timestamp value and a second timestamp value associated with a subsequent entry in the subset of log files exceeds a timeout value" as the log file contains lines that are either directives or entries. An entry corresponds to a single HTTP transaction and consists of a sequence of fields (e.g., integer, fixed, URI, date, time, and string). The meaning of the fields in an entry is specified by a field directive specified in the log file (column 9, lines 12-25).

Also, Carothers teaches in order to minimize the effect of customer time-outs, for valid sessions that end due to a customer timeout, the session length of each of those sessions is reduced by a predetermined time, such as 6 minutes (the standard session time out value). The time-out value is configurable by the business (page 9, paragraph 0120). Sessions w/timeout-percent of all is the number of valid customer sessions that ended due to a session time out as a percentage of all valid sessions; sessions w/other end-count is the number of valid customer sessions that ended due to some reason other than a customer sign-off or timeout (page 10, paragraph 0125).

As to claim 6, Nareddy does not explicitly teach the claimed limitation "outputting all entries in the subset of log file entries that do not belong to a complete client session as raw log data".

Carothers teaches the definition of attempted transactions includes both successfully completed and unsuccessfully completed transactions. Successful transactions mean those home banking system functions/transactions that customers have successfully completed. Each successful transaction has a transaction completion activity log record with Completion Status Code=0 (Transaction successfully completed). Failed transactions mean transactions that have Completion Status Code (page 8, paragraph 0117; see also figure 14).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy and Carothers before him/her, to modify Nareddy log file entries that do not belong to a complete client session as raw log data because that would provide a method for managing information concerning financial transactions for home banking and develops messages which go into the database easily and efficiently as taught by Carothers (page 1, paragraph 0004).

As to claim 7, Nareddy does not explicitly teach the claimed limitation "outputting as raw log data all entries in the subset of log file entries that belong to an incomplete client session which has a beginning entry but no end entry".

Carothers teaches successful financial transactions is the number of financial transactions that were attempted and completed successfully; successful non-financial transactions is the number of non-financial transactions that were attempted and completed successfully; and failed transactions is the number of both financial and non-financial transactions that were attempted but did not complete successfully (page 9, paragraph 0120).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy and Carothers before him/her, to modify Nareddy log file entries that belong to an incomplete client session because that would provide a method for managing information concerning financial transactions for home banking and develops messages which go into the database easily and efficiently as taught by Carothers (page 1, paragraph 0004).

As to claim 11, Nareddy teaches the claimed limitations:

"A system for session-based processing of log files using a data processing system and network session data collected from one or more users, the system comprising a log file collection system for collecting a plurality of server request entries, wherein a server request entry comprises a session identifier; a processing engine to process a subset of the plurality of server request entries to group the server request entries by session using the session identifier in each server request entry" as the data processor component includes a parser, data storage area, and a loader. The data processor component inputs parser configuration data and a log file and updates the main data warehouse. The parser configuration data may include a mapping of actual web sites to logical sites and a mapping of a combination of Uniform Resource Identifiers (URIs) and query strings of the log entries to page definitions (e.g., categories) and event definitions. The parser processes the entries of the log file to generate facts and dimensions to eventually be stored in the main data warehouse. The parser identifies events in accordance with the parser configuration data. The parser includes a filter log entry component, a normalize log entry component, a generate dimensions component, an identify sessions component, and generate aggregate statistics component. The filter log entry component identifies which log entries should not be included in the main data warehouse. For example, a log entry that has an invalid format should not be included. The normalize log entry component normalizes the data in a log entry (column 8, lines 26-49; see also figure 3).

the contained information for each entry further includes information related to a manner of identifying a web site server that responded to the request for that entry, wherein each of the interaction type definitions is associated with a logical site definition that specifies a manner of identifying a web site server related to the web site and that specifies times when that logical site definition is applicable (claim 29).

"the server request entries by session" as when the data warehouse server receives customer data, it converts the customer data into a format that is more conducive to processing by decision support system applications used to analyze customer data. For example, the data warehouse server may analyze low-level navigation events (e.g., each HTTP request that is received by the customer web site) to identify high-level events (e.g., a user session). The data warehouse server then stores the converted data into a data warehouse (column 5, lines 53-60; element 603 of figure 6, filter by IP address (e.g., group the server request by server IP address); see also column 25 table 3, server ID=IP address 209.67.55.102).

Nareddy does not explicitly teach the claimed limitation "group" the server request entries by session.

Carothers teaches each transaction entry is examined to determine a type of function for the transaction, each transaction encountered in the examination is written as a line to an output file, and all transactions encountered are grouped by transaction according to a data column of the transaction journal record that refers back to a session log record. The unique integer key value is assigned to each individual transaction record in the ASCII text output files and written to a pre-defined column in each output file for loading to the relational database system. In addition, transaction times are computed for each of the individual transaction records and added to the corresponding output file (page 1, paragraph 0010; see also figure 15, Session Summary Report).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy and Carothers before him/her, to modify Nareddy group the server request entries by session because that would provide a method for managing information concerning financial transactions for home banking and develops messages which go into the database easily and efficiently as taught by Carothers (page 1, paragraph 0004).

As to claim 13, Nareddy teaches the claimed limitations:

"The processing engine uses a sliding memory window to process the subset of the plurality of web server request entries" as this web page is shown displayed within a web browser display window. The displayed web page includes multiple frames that are each able to display different content, including a control frame with various user-selectable controls and display frames in which customer-specific information is displayed. In the illustrated embodiment, the URL indication corresponds to the information displayed in the display frames. The path portion of the indicated URL specifies an executable Active Server Page program on the server that will supply the content displayed in the display frames, and the indicated URL also includes a query string portion that will be supplied as input to the executable program (column 21, lines 30-40).

As to claim 14, Nareddy teaches the claimed limitations:

"A parser for further analysis the web server request entries that have been grouped by session to generate a user session history" as web sites typically collect extensive information on how its users use the site's web pages. This information may include a complete history of each HTTP request received by and each HTTP response sent by the web site. The web site may store this information in a navigation file, also referred to as a log file or click stream file (column 3, lines 25-35).

As to claim 15, Nareddy teaches the claimed limitations:

"the processing engine generates an output file containing web server request entries corresponding to one or more complete user sessions" as the higher level category of a web page may be identified using a mapping of web page URIs to categories. These categories may be stored in a category dimension table. Also, certain facts, such as the collection of log entries that comprise a single user web access session or visit, may only is derivable by analyzing a series of log entries (column 6, lines 29-36).

As to claim 16, Nareddy does not explicitly teach the claimed limitation “the processing engine generates an output file containing web server request entries corresponding to one or more incomplete user sessions”.

Carothers teaches successful non-financial transactions is the number of non-financial transactions that were attempted and completed successfully; and failed transactions is the number of both financial and non-financial transactions that were attempted but did not complete successfully (page 9, paragraph 0120).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy and Carothers before him/her, to modify Nareddy log file entries that belong to an incomplete client session because that would provide a method for managing information concerning financial transactions for home banking and develops messages which go into the database easily and efficiently as taught by Carothers (page 1, paragraph 0004).

As to claim 17, Nareddy does not explicitly teach the claimed limitation “the processing engine generates an output file containing web server request entries corresponding to one or more user sessions that do not include an end session entry”.

Carothers teaches the definition of attempted transactions includes both successfully completed and unsuccessfully completed transactions. Failed transactions mean transactions that have Completion Status Code (page 8, paragraph 0117; see also figure 14).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy and Carothers before him/her, to modify Nareddy log file entries that do not belong to a complete client session as raw log data because that would provide a method for managing information concerning financial transactions for home banking and develops messages which go into the database easily and efficiently as taught by Carothers (page 1, paragraph 0004).

As to claim 18, Nareddy teaches the claimed limitations:

“A system for parsing web site logs one session at a time, comprising: means for storing network session data from at least one server log file” as a system for providing customers with access to and analysis of interaction or usage data (e.g., navigation data collected at customer web sites or computer program event information) is provided. A data warehouse system collects customer data from the customer web sites and stores the data at a data warehouse server. The customer data may include application event data (e.g., click stream log files), user attribute data of users of the customer web site (column 5, lines 26-31).

“Means for reading a subset of the network session data” as the subset of Internet sites that comprise the World Wide Web network also supports a standard protocol for requesting and receiving web page documents (column 1, lines 44-46).

“Means for processing the subset of the network session data” as a user can request a particular resource (e.g., a web page or a file) that is available from a web server by specifying a unique URI for that resource, a URI can be a URL, Uniform

Resource Name (URN), or any other formatted string that identifies a network resource (column 1, lines 59-67). The data warehouse may include fact tables and dimension tables that represent high-level facts and attributes derived from the low-level facts and attributes of the log files. For example, the higher level category of a web page may be identified using a mapping of web page URIs to categories. These categories may be stored in a category dimension table. Also, certain facts, such as the collection of log entries that comprise a single user web access session or visit (e.g., group by network session), may only be derivable by analyzing a series of log entries (column 6, lines 25-36). The data viewer may be a web browser; the data warehouse server includes a data receiver component, the data processor component, the data warehouse, and decision support system applications. The data receiver component receives customer data sent by the data collection components executing at the various customer web sites. The data processor component processes the customer data and stores it in the data warehouse. The decision support system application provides the customer with tools for analyzing and reviewing the customer data that is stored in the main data warehouse (column 7, lines 7-17).

"Means for generating a first output file containing network session data" as the customer servers and the data warehouse server are interconnected via the Internet. Customer components executing on a customer server includes a data collection component and a data viewer. The data viewer may reside on a client computer of the customer, rather than a server. The data collection component collects the customer data from the storage devices of the customer servers. The data viewer provides access for viewing of data generated by the decision support system applications of the data warehouse server (column 7, lines 3-12).

"Means for parsing said first output file" as the data processor component may have a parser component and a loader component. The parser of the data processor parses and analyzes a log file and stores the resulting data in a local data warehouse that contains information for only that log file (column 6, lines 5-10).

Nareddy does not explicitly teach the claimed limitation "grouped by session".

Carothers teaches each transaction entry is examined to determine a type of function for the transaction, each transaction encountered in the examination is written as a line to an output file, and all transactions encountered are grouped by transaction according to a data column of the transaction journal record that refers back to a session log record (page 1, paragraph 0010; see also figure 15, Session Summary Report).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy and Carothers before him/her, to modify Nareddy group by session because that would provide a method for managing information concerning financial transactions for home banking and develops messages which go into the database easily and efficiently as taught by Carothers (page 1, paragraph 0004).

As to claim 19, Nareddy teaches the claimed limitations:

"Means for reading a subset of the network session data comprises a sliding window" as this web page is shown displayed within a web browser display window.

The displayed web page includes multiple frames that are each able to display different content, including a control frame with various user-selectable controls and display frames in which customer-specific information is displayed. In the illustrated embodiment, the URL indication corresponds to the information displayed in the display frames. The path portion of the indicated URL specifies an executable Active Server Page program on the server that will supply the content displayed in the display frames, and the indicated URL also includes a query string portion that will be supplied as input to the executable program (column 21, lines 30-40).

5. Claims 8-10, 12 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nareddy et al. (US Patent No. 7,107,338 B1) in view of in view of Carothers et al. (US Patent Application No. 2002/0016771 A1) and Dow et al. (US Patent Application No. 2004/0221311 A1, hereinafter "Dow").

As to claim 8, Nareddy teaches the claimed limitations:

"An article of manufacture having at least one recordable medium having stored thereon executable instructions and data which, when executed by at least one processing device, cause the at least one processing device" as a method, system and computer-readable medium for analyzing interaction or usage data, such as for customers (abstract). The data warehouse system includes customer components that execute on the customer servers and data warehouse components that execute on the data warehouse server. The customer servers and the data warehouse server are interconnected via the Internet (column 7, lines 1-5).

"Read a plurality of records from a file system into a buffer, where said plurality or records comprises a subset of all records in the file system" as while various data parsing information and other information is being stored before being used, these items or portions of them can be transferred between memory and other storage devices for purposes of memory management and data integrity, data and data structures may also be stored on a computer-readable medium to be read by an appropriate drive (column 46, lines 33-45). The retrieved category type definition information may already be separated into separate version groups, if it is possible to determine from the information received a subset of the version groups will apply to all of the log entries in the log file (column 49, lines 55-58).

"Scan each record in the buffer to identify a user session for said record and to identify any start or end records in the buffer; allocate, for each identified user session" as the parser includes a filter log entry component, a normalize log entry component, an identify sessions component, and generate aggregate statistics component. The filter log entry component identifies which log entries should not be included in the main data warehouse. The identify sessions component processes the parsed log file data stored in the local data warehouse to identify user sessions. The generate aggregate statistics component aggregates data for the log file being processed as each log entry is processed or after the log file is parsed. The data storage area includes a local data warehouse which is stored non-persistently (i.e., buffer) in main memory of the computer system (column 8, lines 26-65). The routine may track the number of log entries that have been filtered out, the routine then loops to select the next log entry, the routine outputs the log file statistics. The routine invokes the identify sessions routine that scans the log entry table to identify the user sessions and updates a session dimension table. The routine invokes the generate aggregate statistics routine to generate various statistics and then completes (column 13, lines 41-50).

FIG. 13 is a flow diagram illustrating the identify sessions routine in one embodiment. This routine scans the log entry table of the local data warehouse to identify user sessions. In one embodiment, a user session may be delimited by a certain period of inactivity (e.g., thirty minutes). The criteria for identifying a session may be configurable on a customer-by-customer basis; the routine selects the next user from the user dimension table. If all the users have already been selected, then the routine returns, else the routine continues, the routine selects the next log entry for the selected user in time order. If all log entries for the selected user have already been selected, then the routine loops to select the next user, if the selected log entry indicates that a new session is starting (e.g., its time is more than 30 minutes greater than that of the last log entry processed), then the routine continues, else the routine loops to select the next log entry for the selected user. The routine updates a session fact table to add an indication of the new session. The routine then loops to select the next log entry for the selected user. The routine may also update the log entries to reference their sessions (column 16, line 62 to column 17, line 3).

“Output the grouped records for further analysis” as the analysis reports will use the information from the previous analysis (e.g., stored information), and in other embodiments the analysis can be dynamically performed when a report is requested by a customer (column 21, lines 62-67; see also figure 25).

Nareddy does not explicitly teach the claimed limitation "a ring buffer; identify any end records in the ring buffer".

Carothers teaches a status is returned to the VB caller, indicating whether the end-of-file (EOF) is reached, or if a garbled message was encountered. If the status returned is OK, DAPARSER can assume the message contents are isolated in a buffer inside the TJPARSE.OCX control (page 6, paragraph 0075; see also element S5 of figure 8). The called method invokes an isolating the next transaction journal message API to position the next transaction message in the transaction journal log, read the transaction message contents, and insert the transaction message in a static buffer (e.g. ring buffer) (page 2, paragraph 0014).

Although Nareddy teaches the routine creates a mapping of the local identifier (e.g., index into the local dimension table) of the selected entry to the main identifier (e.g., index into the main dimension table) for that selected entry (column 17, line 65 to column 18, line 3).

Nareddy does not explicitly teach the claimed limitation “an index to identify all records in the ring buffer that are associated with the identified user session and to identify all start or end records; and process the index to group all records in the ring buffer belonging to a complete user session”.

Dow teaches the event buffer module receives and aggregates events from an event index. In one embodiment, the event buffer receives selected events from the content navigator module that it has read from an index file. The event buffer module stores a plurality of events for analysis. In one embodiment, the event buffer module buffers event data for a set period of time, such as 2 minutes. For example, the event buffer module may include a 2-minute ring buffer. In an alternate embodiment, the event buffer module buffers event data for an entire video program. In still another embodiment, the event buffer module stores a set number of events before the earliest stored event data is pushed off the event stack to make room for a new event. The other modules in the group detector module analyze the data stored by the event buffer module (page 14, paragraph 0112).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy, Carothers and Dow before him/her, to modify Nareddy process the index to group all records in the ring buffer because that would provide improved navigation of video content recorded from a television broadcast and effective in quickly navigating through commercials as taught by Dow (page 2, paragraph 0010-0011) or use ring buffer technology provide easily and efficiently managing information concerning financial transactions as taught by Carothers (page 1, paragraph 0004).

As to claim 9, Nareddy teaches the claimed limitations:

“the index comprises: a session record for each identified user session for keying into the ring buffer to identify log records associated with said identified user session” as When the data warehouse server receives customer data, it converts the customer data into a format that is more conducive to processing by decision support system applications used to analyze customer data. For example, the data warehouse server may analyze low-level navigation events (e.g., each HTTP request that is received by the customer web site) to identify high-level events (e.g., a user session). The data warehouse server then stores the converted data into a data warehouse (column 5, lines 53-66).

"a hash table for keying into the session record based upon session key information; a linked listing of last seen log records for each session; and a linked list of first seen log records for each session" as the higher level category of a web page may be identified using a mapping of web page URIs to categories. These categories may be stored in a category dimension table. Also, certain facts, such as the collection of log entries that comprise a single user web access session or visit, may only is derivable by analyzing a series of log entries (column 6, line 30-37). Each entry in a dimension table includes the attribute value (e.g., user identifier) and a hash value (column 14, lines 57-61).

Each category includes a unique ID listed that corresponds to the IDs listed in table. In addition, hierarchy information for the categories is provided via column of table, in which each category can optionally have the ID of another category listed as its parent category (column 47, lines 45-55). If the content set is a web site, processing begins at the home web page for the web site, and the various links on the web pages of the web site are variously followed (or crawled) to identify all of the available web pages and the relationships indicating what web pages have links to what other web pages (column 51, lines 39-47).

As to claim 10, Nareddy teaches the claimed limitations:

“the ring buffer implements a sliding window to process all of the log records in the file system into complete user sessions by sequentially adding and removing log records to the ring buffer until all of the log records in the file system have been processed” as this web page is shown displayed within a web browser display window. The displayed web page includes multiple frames that are each able to display different content, including a control frame with various user-selectable controls and display frames in which customer-specific information is displayed. In the illustrated embodiment, the URL indication corresponds to the information displayed in the display frames. The path portion of the indicated URL specifies an executable Active Server Page program on the server that will supply the content displayed in the display frames, and the indicated URL also includes a query string portion that will be supplied as input to the executable program (column 21, lines 30-40).

Nareddy does not explicitly teach the claimed limitation “the ring buffer”.

Dow teaches the event buffer module receives and aggregates events from an event index. In one embodiment, the event buffer receives selected events from the content navigator module that it has read from an index file. The event buffer module stores a plurality of events for analysis. In one embodiment, the event buffer module buffers event data for a set period of time, such as 2 minutes. For example, the event buffer module may include a 2-minute ring buffer (page 14, paragraph 0112).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy, Carothers and Dow before him/her, to modify Nareddy all records in the ring buffer belonging to a complete user session because that would provide improved navigation of video content recorded from a television broadcast and effective in quickly navigating through commercials as taught by Dow (page 2, paragraph 0010-0011).

As to claim 12, Nareddy teaches the claimed limitations:

"the processing engine uses a plurality of data structures to group the web server request entries by session, said plurality of data structures comprising: a per-session record for keying into the ring buffer, a hash table for keying into the per-session records, a linked list of last processed web server request entries for each session, and a linked list of first processed web server request entries for each session" as the parser component may separate the data parsing information entries into separate version groups before processing of the log entries. In addition, new versions of data parsing information can be used for reasons other than changes to a web site or other content set, such as a change in event types or category types of interest to a customer (column 44, lines 60-67). These categories may be stored in a category dimension table. Also, certain facts, such as the collection of log entries that comprise a single user web access session or visit, may only is derivable by analyzing a series of log entries (column 6, line 30-37). Each entry in a dimension table includes the attribute value (e.g., user identifier) and a hash value (column 14, lines 57-61). This web page is shown displayed within a web browser display window. The displayed web page includes multiple frames that are each able to display different content (column 21, lines 30-40).

Nareddy does not explicitly teach the claimed limitation "A ring buffer for storing the subset of the plurality of web server request entries".

Dow teaches the event buffer module receives and aggregates events from an event index. In one embodiment, the event buffer receives selected events from the content navigator module that it has read from an index file. The event buffer module stores a plurality of events for analysis. In one embodiment, the event buffer module buffers event data for a set period of time, such as 2 minutes. For example, the event buffer module may include a 2-minute ring buffer (page 14, paragraph 0112).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy, Carothers and Dow before him/her, to modify Nareddy all records in the ring buffer belonging to a complete user session because that would provide improved navigation of video content recorded from a television broadcast and effective in quickly navigating through commercials as taught by Dow (page 2, paragraph 0010-0011).

As to claim 20, Nareddy does not explicitly teach the claimed limitation "Means for reading a subset of the network session data comprises a ring buffer".

Dow teaches the event buffer module receives and aggregates events from an event index. In one embodiment, the event buffer receives selected events from the content navigator module that it has read from an index file. The event buffer module stores a plurality of events for analysis. In one embodiment, the event buffer module buffers event data for a set period of time, such as 2 minutes. For example, the event buffer module may include a 2-minute ring buffer (page 14, paragraph 0112).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made, having the teachings of Nareddy, Carothers and Dow before him/her, to modify Nareddy all records in the ring buffer belonging to a complete user session because that would provide improved navigation of video content recorded from a television broadcast and effective in quickly navigating through commercials as taught by Dow (page 2, paragraph 0010-0011).

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to James Hwa whose telephone number is 571-270-1285. The examiner can normally be reached on 8:00 – 5:00. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Don Wong can be reached on 571-272-1834. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR.
Status information for unpublished applications is available through Private PAIR only, for more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the PAIR system contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

08/20/2009

/James Hwa/

Examiner, Art Unit 2163

/Wilson Lee/

Primary Examiner, Art Unit 2163